



CLAIMS:

1. We claim a method for creating a well-defined architecture that combines configurations at the network, server and storage tier of an infrastructure in order to provide for scalability of services incrementally, increased performance, and enhanced security made up of the following configuration tasks:
 - 5 (a) Configuring several Virtual Local Area Network (VLAN) segments in order to separate traffic from server to disk, from content switch to server, from user or WAN connected ethernet segment to content switch.
 - 10 (b) Configuration of router access lists such that traffic is protected across the above mentioned VLAN segments.
 - (c) Configuration of many redundant inexpensive server machines.
 - (d) Above mentioned machines are configured exactly the same (memory, CPU, disk).
 - (e) Each server contains the exact software image of all other servers and machine dependent configurations are stored in LDAP.
 - 15 (f) Configuration of Network Attached Storage (NAS) technology along with Network File System (NFS) such that machines can share storage and file locking is managed via NFS.
 - (g) This storage is configured in a Redundant Array of Inexpensive Disk (RAID) configuration.
2. We claim that in order to make use of the method in claim 1, method for grouping or clustering software together is described as follows:
 - (a) Each server mentioned in claim 1 will have an exact copy of the complete software grouping.
 - (b) The software grouping consists of an Email MTA, Web-based email front-end, POP daemon, IMAP daemon, Chat daemon, Web daemon, backup server software, monitoring daemon and agents, Web-based content portal, and additional software as it becomes useful to users of service providerships.
 - (c) Users of the system will be directed to the least loaded and most available server by way of a content switch.
 - (d) Any server will be able to handle the user request for service or software application.
 - (e) Software can be added to the grouping at any time after it has been through a full level of quality assurance testing.
 - 30 (f) After new software, bug fixes or security updates have been through a full level of quality assurance testing, they can automatically be pushed to the hardware devices within the architecture defined by claim 1.
 - (g) A Lightweight Directory Access Protocol (LDAP) configuration database will store any independent server configurations that will identify slight differences in the software.
 - 35 (h) Above mentioned LDAP configuration database will not impact the software grouping at all, but will serve to extend the grouping.

3. We claim that the methods in claims 1 and 2 will significantly decrease cost, increase scalability, increase redundancy and enhance security.